Hindawi
Mobile Information Systems
Volume 2021, Article ID 6644389, 12 pages
https://doi.org/10.1155/2021/6644389

Research Article

Event Driven Duty Cycling with Reinforcement Learning and Monte Carlo Technique for Wireless Network

Han Yao Huang, Tae-Jin Lee, and Hee Yong Youn

College of Information and Communication Engineering, Sungkyunkwan University Suwon, Seoul, Republic of Korea

Correspondence should be addressed to Hee Yong Youn; ahczhg@skku.edu

Received 26 October 2020; Revised 3 February 2021; Accepted 22 February 2021; Published 8 March 2021

Academic Editor: Marco Anisetti

Reducing transmission delay and maximizing the network lifetime are important issues for wireless sensor networks (WSN). The existing approaches commonly let the nodes periodically sleep to minimize energy consumption, which adversely increases packet forwarding latency. In this study, a novel scheme is proposed, which effectively determines the duty cycle of the nodes and packet forwarding path according to the network condition by employing the event-based mechanism and reinforcement learning technique. This allows low-latency energy-efficient scheduling and reduces the transmission collision between the nodes on the path. The Monte Carlo evaluation method is also adopted to minimize the overhead of the computation of each node in making the decision. Computer simulation reveals that the proposed scheme significantly improves end-to-end latency, waiting time, packet delivery ratio, and energy efficiency compared to the existing schemes including S-MAC and event-driven adaptive duty cycling scheme.

1. Introduction

Wireless Sensor Network (WSN) has been used for a wide range of applications, primarily for target area monitoring [1]. Event monitoring applications such as intrusion, lightning, or fire detection should be designed according to their operating condition [2]. In WSN, a large number of sensor nodes are distributed in the target area, which can process the signal and communicate with each other. The major problem in such a WSN-based monitoring system is the limited energy of the nodes, and, therefore, it is important to minimize the energy consumption of these for extensive network operation. Various energy-efficient communication algorithms and schemes have been proposed to maximize the life of the WSN. The Media Access Control (MAC) layer is responsible for scheduling nodes in WSN to effectively manage communication between nodes.

The method commonly adopted with the MAC protocol for minimizing energy consumption in WSN is duty cycling. Here, the nodes stay awake only a fraction of time for sensing and communication. The periodic dormancy, however, increases the transmission delay, which is detrimental especially to human life-critical applications. Energy-saving at the sacrifice of performance might be fatal for them. The transmission delay is caused by the sleeping nodes on the multihop path between the source and the destination node, called sleep latency [2–7]. This is a serious concern with WSN where the transmission range of a node is usually smaller than the distance between the communicating nodes. As the network operation is dynamic, the duty cycle of the nodes is required to be continuously adapted to avoid early sleep under high traffic load or overlistening under low traffic load. Event-driven adaptive duty cycling of the nodes can satisfy this requirement, which is the main objective of this paper.

It was shown that a significant amount of energy can be saved by employing sleep and idle listening mode for the nodes [8]. The duty cycle-based MAC protocols are classified into synchronous and asynchronous approaches. In the synchronous protocol, such as S-MAC [7], T-MAC [9], RMAC [8], and P-MAC [10], a schedule table is created for all the nodes to specify the sleep and wake-up time. S-MAC is based on broadcasting the preframe of SYNC and DATA packet for scheduling. Here, the performance metrics related to the network operators were not included in designing the protocol [10]. The asynchronous MAC protocol such as B-MAC [11], X-MAC [12], and RI-MAC [13] allows the nodes to operate independently to enhance the adaptability against dynamic load changes. To achieve a more adaptive schedule, the authors of [14] have shown that a significant amount of energy can be saved, and the delay is reduced by dynamically adjusting the latency. BADCS is proposed to reduce event detection latency and data routing delay using a duty cycle adjustment algorithm [15].

In this paper, a novel event-driven scheduling approach employing the reinforcement learning (RL) algorithm is proposed to reduce the sleep latency and improve the performance of packet switching in WSN. It adjusts the duty cycle of the nodes in the multihop path according to the status of the network so that the delay and waiting time incurred during packet transmission can be minimized. Here, the low-delay energy-efficient transmission path from the source to the sink node is decided using the RL algorithm. For a node on the path, the feedback information on the delay and energy taken by the path is provided to its next-hop nodes called the parent nodes. The RL algorithm is used to choose the best parent node and wake it up for forwarding the data. Additionally, to reduce the waiting time due to early sleep, the node of high traffic such as the one having many neighbors or close to the sink node is woken up for a relatively long time. The simulation results show that the proposed approach substantially outperforms S-MAC and the existing adaptive duty cycling scheme [16] under various network conditions. The main contributions of the paper are summarized as follows:

(i) The existing node scheduling problem is transformed into a decision problem employing the event-driven approach and RL to effectively deal with the dynamically changing network condition of WSN. The transmission delay is due to early sleep and transmission collision. Early sleep is avoided by the event-driven approach to wake up the sleeping nodes promptly, while transmission collision is avoided by the RL technique to properly select the forwarding path.

(ii) The existing MAC protocols are based on local feedback information in deciding the schedule. In this paper, the Monte Carlo (MC) evaluation technique is employed to obtain global information and sampling, which greatly improves the speed and accuracy for finding a suitable schedule.

(iii) A technique for finding maximum achievable reward in RL is developed by solving Bellman's optimal equation, which allows accurate solutions in the small number of computation steps.

The rest of the paper is organized as follows: in Section 2, the work related to duty cycling and RL-based scheduling for the MAC of WSN is discussed. The proposed scheme is presented in Section 3. Section 4 discusses the simulation results, and the conclusion is made in Section 5.

2. Related Work

2.1. Duty Cycling. Generally speaking, each sensor node in WSN operates on battery power, where two factors affect the rate of energy consumption. Firstly, the rate is high if the transceiver is in transmission, reception, idle (or overhearing), and low during sleeping. Secondly, the event other than successful packet transmissions such as collision or retransmission causes energy waste. Also, the existence of two kinds of delays explained below increases the transmission time, which is affected by transmission characteristics and duty cycle.

> *Early Sleep Delay.* Assume that some packets in a node are needed to be sent to another node that awakes and sleeps periodically. The problem with early sleep occurs when a packet is sent to the sleeping node on the multihop path, and the data transmission is delayed until it switches back to the active state.
>
> *Transmission Collision Delay.* Collision occurs if some nodes send packets at the same time when they are in the transmission range of the other node.

Figure 1 compares two types of duty cycling schemes. As shown in Figure 1(a), the nodes of S-MAC periodically switch from sleep to listen mode for prolonging the lifetime. Only the nodes in the listen mode can receive, forward, or process the packets. If the packet arrives during the sleep mode (event-A of Figure 1(a)), the process is delayed until the node switches to the listen mode. Therefore, the latency with periodic duty cycling is usually high. Figure 1(b) shows event-driven duty cycling, which controls the listen/sleep mode of a node based on the arrival and departure event of a packet [16]. Here, the next-hop node is woken up when a packet arrives to reduce the latency.

Various event-driven approaches have been proposed to address the problem of delay caused by early sleep [17], and the state change of a node is promptly reported by continuous monitoring of the operation. While the event-driven approach reduces the transmission time and energy consumption of a node, an efficient scheme needs to be developed to properly reflect the occurrence of the events to the scheduling. The machine learning technique such as RL is effective for meeting this requirement. RL is a biology-based machine learning approach that acquires knowledge by exploring the operation environment without external supervision or prior knowledge. Numerous studies have been conducted on RL for various applications [18–21], including the reduction of transmission delay and maximization of sensor node lifetime [22, 23]. Improving the performance of the network by replacing time-based duty cycling with event-driven reinforcement learning (EDRL) is the main objective of this paper.

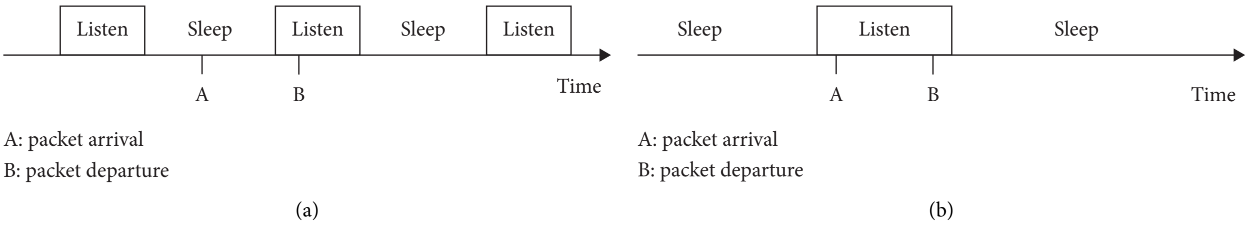


Figure 1: The comparison of two duty cycling schemes.

2.2. Markov Decision Process (MDP). MDP is an analytical model applicable to the process having Markov property, modeled by 4 tuples <*S*, *A*, *R*, *P*> defined below:

(i) S: a finite set of states, where s_i is the state at step-i

(ii) A: a finite set of actions.

(iii) $P(a|s_n, s_{n+1})$: the probability that action a, leads the system in s_n to $s_{(n+1)}$. $S \times A \times S \longrightarrow [0, 1]$ is the state transition probability density function.

(iv) $R(a|s_n, s_{n+1})$: the return after the transition from s_n to $s_{(n+1)}$ due to action a. $S \times A \longrightarrow R$ is the reward function.

A key feature of MDP is the Markovian property; the probability to reach state s at step-n depends on only the previous step, step-$(n-1)$ [24]. In discrete-time MDP, which is considered in this paper, the agent is in state $s_n (\in S)$ and takes action $a (\in A)$ according to the policy, π, at step-n. In response to the action, the environment provides scalar feedback, called a reward, $R(a|s_n, s_{(n+1)})$. This process is illustrated in Figure 2, where the value of state $v(s_n)$ and action, $q(s, a)$, is returned as the reward. RL is a commonly employed solution for MDP when the application possesses the Markov property. RL algorithm aims to find a policy that maximizes the accumulated reward. If the system operates in a finite time domain, it can be solved using the dynamic programming approach and Bellman optimality equation. Otherwise, it is solved using the value iteration, policy iteration, linear programming, approximation method, or online learning technique [25]. RL has been used to solve the typical sequence decision problem, using the learner and decision-maker called agent [26]. The agent chooses a good action based on only the current sensory observation and remembers the past sensations to select a good action [27]. The proposed scheme is presented next.

3. The Proposed Scheme

In this section, the proposed scheme is presented, which decides the communication path using RL, which minimizes the transmission delay and energy consumption. The list of notations used in the paper is given in Table 1.

3.1. Design Goal. Regarding packet transmission in WSN, the transmission delay and energy efficiency are conflicting factors due to the limited energy of the nodes. Various protocols have been developed to reduce the transmission delay between the nodes of finite energy. The primary task of WSN is to monitor and report abnormal or emergency conditions, and each node in the network may serve as a source or relay node. The existence of a duty cycle increases the delay due to early sleep. Another cause of delay is a collision. The proposed scheme effectively avoids early sleep by employing an event-driven approach to wake up the sleeping nodes promptly and avoids transmission collision by the RL technique properly selecting the forward path.

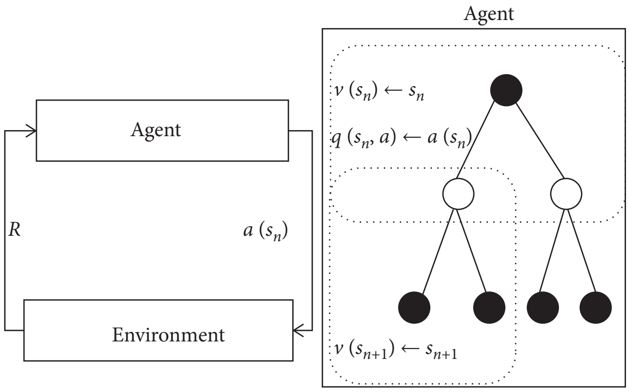

Agent
Agent
Environment
R
$a(s_n)$
$v(s_n) \leftarrow s_n$
$q(s_n, a) \leftarrow a(s_n)$
$v(s_{n+1}) \leftarrow s_{n+1}$

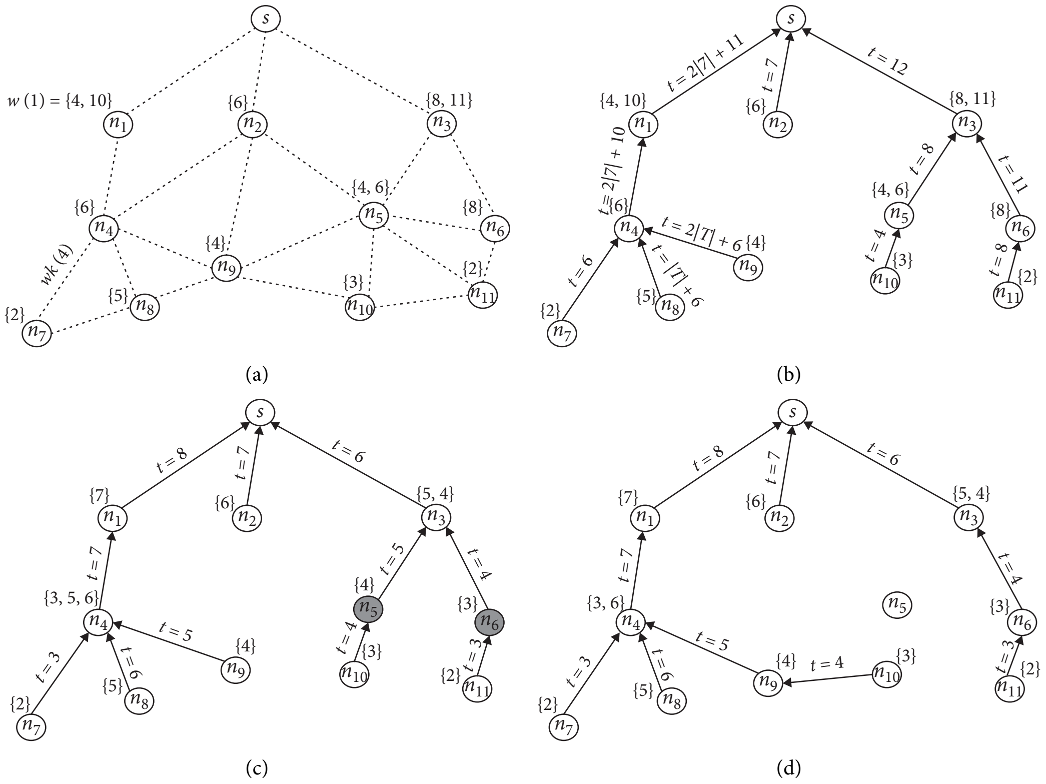
Figure 2: The operational structure of RL.

Considering the trade-off between performance and scalability, an event-based wake-up strategy is adopted. The proposed scheme consists of two phases: *RL phase* and *report* phase. During the RL phase, the nodes of the forwarding path are selected by carrying out exploration producing the consumed energy and delay data as a reward. In the report phase, the value of the RL function is obtained, where the state is input and the state-action pair is output. Then, the function is used to decide and explore the next action with a greedy algorithm. Finally, through the interaction between the nodes and the environment, the optimal wake-up schedule is decided. In the learning process of the proposed scheme, each node selects the forwarding path and then calculates the reward. The result affects the decision and exploration of the next state. Applying the proposed scheme, a proper duty cycle is obtained using the wake-up mechanism for timely transmission. Figure 3 compares the operations of different duty cycling schemes, where the length of the working cycle, $|T|$, is 12 and the number in the bracket denotes active time slots of each node.

Figure 3(a) shows the fixed duty cycle scheme, where the node is woken up on fixed time slots indicated by the number in the bracket. In this case, a dormant node only switches to the active state when (i) it is scheduled to switch

Table 1: The notations used in the paper.

Notation	Description
q_i	The capacity of the queue for nodes as node-i ($i = 1, ..., N$)
S, A, P, R	Components of MDP: state space, action space, transition probability, the reward function
α	Learning rate
γ	Discount factor
$v(i)$	Value of node-i
$G = (V, E)$	WSN with the set of nodes, V, and edges, E
r	Transmission range of a node
$NB(i)$	Neighbor nodes of node-i
$w(i)$	Duration of slots when node-i works
$wk(i)$	Slot when node-i is wake-up
$p(i)$	The parent node of node-i
$c(i)$	The child node of node-i
sch(i)	Transmission schedule of node-i
$F(i)$	Nodes of $NB(i)$ forbidden to wake up
$p_c(i)$	Candidate parent nodes of node-i
$\tau = (n_s, \ldots, n_d)$	The path from the source to the destination node



(a)

(b)

(c)

(d)

Figure 3: An example of comparing different schemes.

to the active state to receive data packets, or (ii) it has some packets to transmit to a receiver that is active at that time. A cycle is divided into 12-time slots, and each is enough to send and receive a packet. In Figure 3(b) of S-MAC, the forwarding nodes and their active slots are predecided and fixed, where n_7 sends data to n_4 at slot-6 because n_4 works only at slot-6 and n_8 has to wait for n_4 to work in the next cycle and send data. The transmission latency is increased due to this problem. In Figure 3(c) of the event-driven scheme, the node wakes up the next-hop node to reduce the waiting delay if it has a packet to transmit. Observe from the figure that n_{10} wakes up n_5 and transmits a packet at slot-4, while n_6 transmits a packet to n_3. Since n_5 is within the transmission range of n_6, a collision occurs causing retransmission, and as a result, the latency becomes greater than 8. As shown in this example, a node needs to be properly chosen when there exists more than one neighbor node to avoid collision and transfer the packet to the sink node fast. Thus, this scheme, waking up appropriate nodes based on reinforcement learning, is proposed to make use of the available time slots and neighbor nodes. The latency can be reduced to 8, as shown in Figure 3(d).

3.2. Operation

3.2.1. Selection of Parent Node. In WSN, $G = (V, E)$, where V is a set of N sensor nodes, and node-i has a queue of the capacity of q_i packets. $E = \{(u, v) | 1 \le u \le N, 1 \le v \le N\}$ denotes the link between node-u and node-v. As in [28], all nodes are assumed to have the same transmission range, r, for simplicity. dis(u, v) $(\in E)$ represents the distance between node-u and node-v, which is smaller than r if node-v is the neighbor node of node-u, i.e., $v \in NB(u)$ $(\text{dis}(u, v) \le r)$. Each node has sleep and work states. Let T denote a work period that is usually divided into a fixed number of time slots. Each slot is long enough so that a source node and a relay node can either cooperatively transmit one data packet to the destination or transmit one of their packets. Then, the work schedule of node-i, $w(i)$, is defined as the active time slots in T. $wk(i)$ is the slot when node-i is woken up and working.

$$wk(i) = c|T| + t_l \, (\in w(i)), \tag{1}$$

where c is a nonnegative integer and t_l is an element in the set of the active time slot of the node. With duty cycling, each node can receive data in only a working state, and thus the time duration for receiving data is quite limited. Concerning the energy efficiency of a single node and lifetime of the entire network, $\min(|wk(i)|)$ and $\min(\sum|wk(i)|(i \in V))$ are the objectives, respectively. The ratio of duty cycle of a node is k/T if it works for k slots. Note that k and $wk(i)$ are fixed with time-based duty cycling. Considering early sleep delay and collision delay, $wk(c(i)) \le wk(i) \le wk(p(i))$ is the basic condition of successful transmission when a packet is transmitted from $c(i)$ to $p(i)$. Here, $p(i)$ and $c(i)$ denote the parent and child node of node-i, respectively, which receives and sends the packet. The delay caused by early sleep is $wk(p(i)) - wk(c(i))$ $(\le |T|)$. The total transmission delay, D, is then

$$D = \sum_{i=1}^{q_i} wk(p(i)) - wk(c(i)) + d_c, \tag{2}$$

where d_c represents the delay caused by transmission collision and duty cycling. The existence of collision between the hidden and exposed node in the wireless network environment causes the delay. The time-based scheduling approach is not efficient due to the synchronization overhead and lack of information on the network condition. The proposed scheme is based on event-driven, and RL helps the nodes make the proper local decision based on the feedback information on the global network status. Here, a mechanism is employed to wake up a proper node on the next hop and alleviate the early sleep problem. As for data transmission scheduling, the goal is to construct a set of collision-free transmission schedules allowing aggregation of the data in the sink node. The delay caused by early sleep is reduced by waking up the node instead of waiting for the termination of sleep period. Note that unreasonable selection of the parent node makes d_c larger. Assume two transmission schedules for node-u and node-v $((u, v) \in E)$, $\{p(u), wk(u)\}$ and $\{p(v), wk(v)\}$. Here, $\{p(u), wk(u)\}$ means that the nodes in the set of the senders of node-u are scheduled to transmit to $p(u)$ at $wk(u)$, which is decided by $c(u)$. A collision-free transmission should satisfy one of the following two conditions:

(1) $wk(v) \ne wk(u)$

(2) $wk(v) = wk(u) \,\&\, p(v) \notin NB(u) \,\&\, p(u) \notin NB(v)$

The transmission schedules are collision-free if the wake-up slots of node-u and node-v are different. Otherwise, their parent nodes must not be in the transmission range of each other ($p(v) \notin NB(u)$ and $p(u) \notin NB(v)$). In the following, the formal definition of the minimum latency problem based on the event-driven scheduling approach is given.

Input:

(1) A duty-cycled sensor network $G = (V, E)$;

(2) A sink node-s.

Output: The schedule, $sch(i)$ $(= \{p(i), wk(i)\}$ $\forall i \in V)$, satisfies the following condition:

(1) $|p(u)| \ge 1$;

(2) $\cup_{i=1}^{m} i = V - \{s\}$;

(3) The length, m, is minimized

(4) Data sent fromni to nj according to sch(i) and sch(j)are collision-free, $\forall i, j \in V \,\&\, i \ne j$;

The wake-up schedule is decided for each node. Here, $wk(i) = wk(c(i)) + \varpi$, while ϖ denotes the time required for transmitting the data from node-$c(i)$. Collision occurs if node-u and node-v send a packet at the same time in the case of $(u, v) \in E$. If $(u, v) \notin E$, it still occurs when $p(v)$ locates inside the transmission range of the other node as $(u, p(v)) \in E$.

To take care of the first cause of collision, for node-u, some nodes are forbidden to be woken at the same time, denoted as $F(u)$. Here, $wk(u)$ is decided by $c(u)$ because the node switches to work state after it is woken up. For the case of $(u, v) \in E$, node-u and node-v are forbidden to be selected as parent node of a node simultaneously.

$$F(u) = \{i \ne p(v) \& i \in NB(p(v)) | \, t = wk(v)\}, \tag{3}$$

For the second cause, a node is forbidden to be selected as a parent node if there is a neighbor node transmitting the packet in the same time slot.

$$F(u) = \{i \ne p(v) \& i \in NB(v) | \, t = wk(v)\}, \tag{4}$$

Combining the two cases, $F(u)$ becomes

$$F(u) = \{i \ne p(v) \& i \in NB(v) \,\&\, i \in NB(p(v)) \, | t = wk(v)\}. \tag{5}$$

The transmission schedule for each node is decided starting from the leaf node while moving toward the sink. At first, the source node decides the schedule for itself and its parent node, and then the data are sent to the parent node, which does the same thing as the child node. The set of nodes, $p_c(i)$ $(=(NB(i) - NB(p(u)) \cup NB(u), (u, i) (\notin E))$, includes the nodes that will cause collision less likely. For $|p_c(i)| > 1$, there exist several neighbor nodes for node-i to be selected as parent node, and thus a weight for each candidate parent node is estimated using RL to select the best one. The selection algorithm of the parent node minimizing the latency is shown in Algorithm 1.

3.2.2. Exploration of Packet Forwarding Path. Assume that neighbor nodes allowing minimum early sleep delay and transmission delay have been selected. Then, a packet is transmitted effectively by minimizing the number of hops, m. The process of obtaining the weight for each transmission schedule is given in the following.

Let $v(i)$ be the state value of node-i obtained from the RL process. A set of nodes forming a path is evaluated using the rewards, and then $v(i)$ is updated. When a new event occurs, node-i having a packet to transmit finds $p(i)$ from $p_c(i)$ and wakes it up for packet forwarding. When the sink node receives data from node-i, it records the path and estimates the reward due to node-i for improving the schedule based on RL.

The following shows the model and the process of solving the target problem using RL. Here, $s(i)$ and $v(i)$ denote state-i and the value of state-i estimated by the RL process, respectively. The state is estimated by the feedback information on the amount of energy consumed and forwarding latency after a packet is successfully transmitted to the succeeding node or sink node. The lower the energy consumption and latency, the larger the estimated value. It is preferred to choose the state of large value when making a decision.

In deciding $v(i)$ using a stochastic decision process, an agent interacting with the environment is implemented in each node. It works as follows. Let $A(s)$ be a finite set of control actions allowed to be taken with a state-space denoted as S such that $s(i)$ $(\in S)$. Suppose that an agent chooses an action, $a_{s(i)}$ $(\in A(s))$, that is available at $s(i)$. After the action, the agent receives an immediate reward, R, and the system makes a transition to a new state, s', $(a_{s(i)} \times p \longrightarrow s')$ with a transition probability, p. Policy π, $\pi(s) \longrightarrow a$, denotes the rule of action selection. An optimal policy, π^*, maximizes or minimizes the objective function. The state of a high value implies that the transition to this state gets more reward. The final solution consists of the states that have a long-term revenue. The value of state-s, $v_\pi(s)$, is defined as a state-value function:

$$v_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right]. \tag{6}$$

Accordingly, the state-action is viewed as a decision made in the current state and evaluated by the value of the state-action. The state-action value function is

$$q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right]. \tag{7}$$

A global track, $\tau^* = \{u_0^*, u_1^*, \ldots, u_m^*\}$, for an m-hop path from node-i to the sink, node-s, is optimal if it satisfies

$$J(\tau^*) = J\{u_0^*, u_1^*, \ldots, u_m^*\} \leq J(\tau), \tag{8}$$

where $J()$ is the object function. u_j $(j \neq i,\ u_j \in V)$, which is decided by node-$(j-1)$, forms a sequence of states minimizing the delay and energy consumption of the whole process. Considering the limited feedback information in the local nodes, it is hard to evaluate the decision once it is made. After making a decision, the node needs to get the feedback on the decision regarding the transmission delay and energy consumption and then update the policy, π. Since they are local information, the Monte Carlo (MC) evaluation technique is employed to obtain global information. The target function, $\int R(\tau) P_\pi(\tau) \mathrm{d}\tau$, is expected value of the cumulative return denoting overall revenue of the policy, π. Define $\eta(\tau, \pi)$ as the average reward of a policy as follows:

$$\eta(\tau,\pi) = \lim_{N \longrightarrow \infty} E_{\pi_0}^{\pi}\left\{\sum_{s_0 \in \tau} r(s_i, s_{i+1})\right\}. \tag{9}$$

Here, $E_{\pi_0}^{\pi}$ is the expectation with the probability measure generated by the policy, π, with initial policy, π_0. RL is used to update π_0 iteratively, leading to the optimal policy, π^*. Maximizing the expected discounted total reward is the objective, which is defined as follows:

$$\max V_\pi(s) = E_{\pi,s}\left[\sum_{t=1}^{T} \gamma^t R(s_t'|s_t), \pi(a_t)\right]. \tag{10}$$

$V_\pi^*(s)$ is the maximum achievable reward at state-s, which is found by solving the following Bellman's optimal equation.

$$V_\pi^*(s) = E_{\pi,s}\left[\sum_{t=1}^{T} \gamma^t R(s_t'|s_t), \pi(a_t)\right]. \tag{11}$$

$v(s)$ and $q(s, a)$ can be obtained using the principle of Bellman optimality:

$$\begin{aligned} v^*(s) &= \max_a R_s^a + \gamma \sum_{s' \in S} p_{ss'}^a v^*(s), \\ q^*(s,a) &= R_s^a + \gamma \sum_{s' \in S} p_{ss'}^a \max_{a'} q^*(s',a'). \end{aligned} \tag{12}$$

Even though the process evolves in the continuous-time domain, a discrete-time model is assumed in this paper, where time is slotted with intervals of unit length. In the proposed scheme, a node is represented as a state.

$$s(i) \times \pi(a|s(i)) \times p \longrightarrow s(u), u \in NB(i), \tag{13}$$

where $s(i)$ indicates that node-i has a packet to transmit, and $s(u)$ is the next hop selected by node-i. The action taken is decided according to π and $s(i)$. E_τ $(\in\{0, 1\})$ equals 1 if an event is reported along the path, τ, and 0, otherwise. $\tau = (n_s, n_i, n_j, \ldots, n_{des})$ represents a path from the source node, n_s, to the destination node, n_{des}. n_τ is the number of nodes in τ. The reward function, $R(s_t, a_t)$, is then given by

$$R(\gamma) = \begin{cases} -\phi\, d - \alpha l, & E_\tau = 1, n_\tau \leq L, \\ -\theta LT, & \text{otherwise}, \end{cases} \tag{14}$$

```
Input: Node-$u$, $NB(u)$ and Node-$v$, $NB(v)$
Output: $p(u)$,$p(v)$
        $p_c(u) \leftarrow NB(u)$, $p_c(v) \leftarrow NB(v)$;
    for $\forall i \in NB(u)$ do
        sch$(u).p(u) \leftarrow \min\{d\}$, $wk(p(u)) \leftarrow$ sch$(u) \cdot wk(u) + 1$;
        $p_c(v) \leftarrow NB(v) - NB(p(v)) \cup NB(v)$
for $\forall j \in p_c(v)$ do
        if $j \in NB(k)$, $|$sch$(k)| > 1$ & $k \in V$ then
        $p_c(v) \leftarrow p_c(v) - \{j\}$;
        if $v(j') \leq v(j)$ then
        sch$(v)$ $p(v) \leftarrow j$;
$wk(p(u)) \leftarrow$ sch$(u) \cdot$ $wk(u) + 1$;
```

Algorithm 1: Selection of parent node for wake-up.

where d and l denote the delay and total energy consumption of the path, respectively. To avoid local optimum solution, the state value is not updated until one period is completed with the MC process, which updates the value as follows:

$$\begin{aligned} V(S_t) &\leftarrow V(S_t) + \alpha(G_t - V(S_t)), \\ G_t &= R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T, \end{aligned} \tag{15}$$

where G_t is the objective of MC. $V(S_t)$ is the expected discounted rewards, which are updated after one path is tried. The method to obtain true expected value by exploring all possible paths is extremely inefficient. Thus, finding an approximate value through effective sampling is a better way. The MC method conducts sufficient sampling of the state space using the ε-soft greedy algorithm.

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{|A(s)|}, & \text{if } a = \operatorname{argmax}_a Q(s,a), \\ \dfrac{\varepsilon}{|A(s)|}, & \text{if } a \neq \operatorname{argmax}_a Q(s,a), \end{cases} \tag{16}$$

$$a_n^* = \arg \max_{a_n \in A(s_n, e_n)} (s_n, s_{n+1} | \theta).$$

For a node having more than one parent node, $|p_c| \geq 2$, there exists $|p_c|^m$ path for the m-hop path. Exploring every path based on RL is not effective. The ε-soft greedy is a popular exploration method used to obtain samples from the probability space and get sampling space, θ, for exploitation. Also, to ensure sufficient and efficient sampling space, the variable, ε, is added to the RL for a better learning process.

The soft greedy policy can ensure sufficient sampling of the state allowing accurate estimation of the state value. The updated state will be relatively small as exploration continues, while excessive exploration delays the convergence to the optimal value. Therefore, a constraint on the update condition for the parameters of the soft greedy policy is needed. ε is used as the constraint. Note that the bigger the change in the state value, the greater the chance of exploring the untried state.

$$\varepsilon = a \frac{\mathrm{d}v(i)}{\mathrm{d}t} + b. \tag{17}$$

The set of sample data, θ, is obtained by the exploration of the environment with the soft greedy policy. The node evaluation process based on RL is shown in Algorithm 2.

The flowchart of the proposed EDRL scheme is depicted in Figure 4. It is implemented in two blocks, the network operation and RL process, which run independently. Unlike the time-based duty cycling, the nodes switch to sleep mode for saving energy until packet transmission is required. The proposed scheme is evaluated next.

4. Performance Evaluation

In this section, the performance of the proposed EDRL scheme is evaluated. It is also compared with S-MAC and the existing adaptive event-driven scheme (ED) [14] in terms of packet delivery ratio, latency, packet loss rate, and energy efficiency as the load varies.

In the simulation, 25 nodes are distributed randomly in a 50 $*$ 50 area, and the nodes send packets to the sink node, via one or multihop path. All the nodes have the same transmission range, and the interference range is equal to the transmission range. Here, one node is selected as the sink node (destination node), which never goes to sleep, while the other nodes periodically generate packets as an event occurs. The parameters used in the simulation are listed in Table 2.

In Figure 5, the delivery ratio of the three schemes is compared as the number of packets per event varies from 50 to 300. Compared to ED and S-MAC, the proposed EDRL scheme yields a consistently higher delivery ratio. Note that the load is high when the number of the packets is large per event. This demonstrates that the proposed scheme is quite effective in dealing with the duty cycle in response to dynamic load change. This is because the decision of the state is made via RL based on the data obtained from the environment, which responds to the changes of the network on time. The selection of the parent node effectively reduces the transmission collision and improves the packet delivery ratio in the network. The soft greedy policy can provide an adequate sampling of the state allowing for an accurate estimate of the state value. Since ε is bigger than the others, however, the performance of EDRL scheme is similar to the other schemes in the beginning.

```
Initialization;
while $e$ is smaller than the number of total episodes do
while $n$ is smaller than the maximum step do
   Take action with $\varepsilon$-soft greedy:
      $\pi(a|s) \leftarrow \begin{cases} 1-\varepsilon+(\varepsilon/|A(s)|), & \text{if } a=\text{argmax}_a Q(s,a), \\ (\varepsilon/|A(s)|), & \text{if } a \neq \text{argmax}_a Q(s,a), \end{cases}$
   While the nodes are in RC phase do
      Wake up the nodes decided from the RL process;
      Generate data packets or receive data packets;
   end while
   Determine the subsequent state;
   $n = n+1$;
Observe the delay and energy consumption;
Compute reward
      $R(\gamma) = \begin{cases} -\phi\, d - \alpha l, & E\tau = 1, n_\tau \le L, \\ -\theta LT, & \text{otherwise}, \end{cases}$
Store transition $(s_n, a_n, s_{n+1}, R)$ and $\tau$ in sample space $Q$;
update $\pi$, $V(s)$, $\varepsilon$;
$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$
$q_\pi(s,a) = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a]$
$e = e + 1$;
end while
```

Algorithm 2: Node evaluation based on RL.

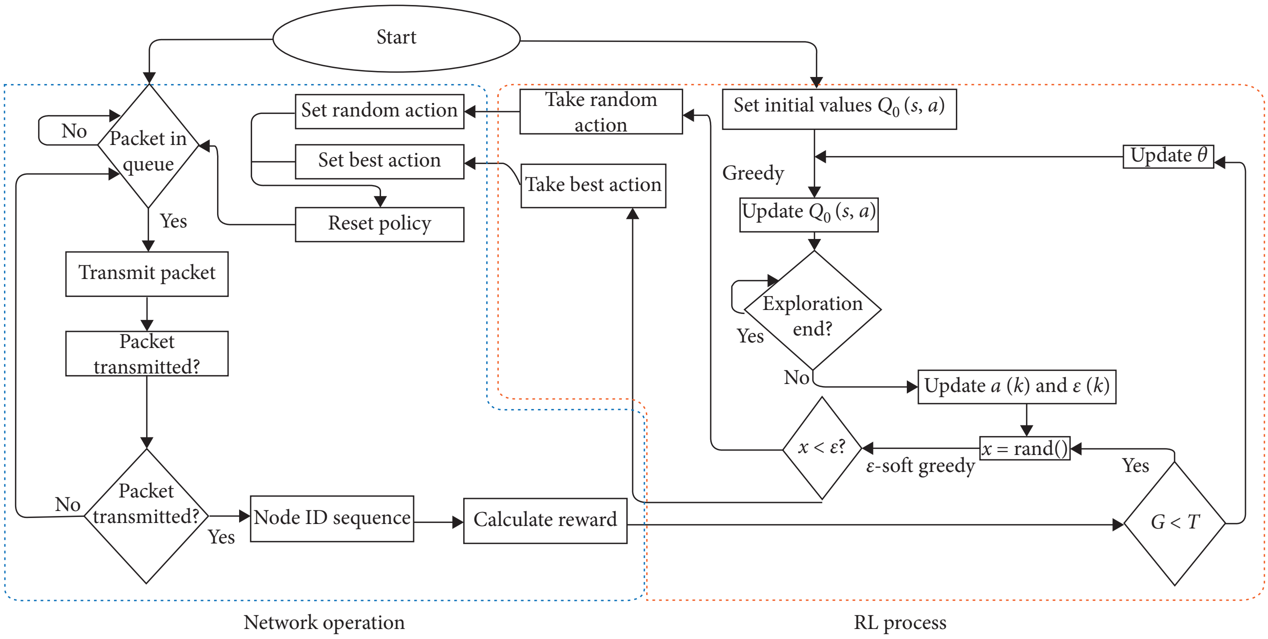


Figure 4: The flowchart of RL the proposed EDRL scheme.

Figure 6 shows the end-to-end latency of packet transmission. It can be observed from the figure that the delay of the proposed EDRL scheme is always smaller than that of the other schemes. S-MAC is relatively insensitive to the load, which indicates that it is not adaptable to the traffic load. The reduction of transmission conflicts and correct path selection make the end-to-end delay smaller compared to the others in the proposed scheme. Small fluctuations with EDRL are due to the RL process involving exploration and periodic MC update of state value and policy.

Figure 7 compares the packet loss rate of the three schemes. Observe from the figure that the ratio with the proposed EDRL is always lower than that of the other schemes regardless of the load. EDRL not only displays the slower rate of packet loss than the others with the increase of the load, but also allows no or little packet loss until the

Table 2: The parameters used in the simulation.

Parameter	Value
Number of nodes	25
Discount factor	0.2
Simulation area	$50 \times 50\,\mathrm{m}^2$
Transmission range	10 m
Learning rate	0.3
Energy consumption in transmitting state	0.66 mW
Energy consumption in receive state	0.395 mW
Energy consumption in idle state	0.350 mW
Energy consumption in sleep state	0 mW

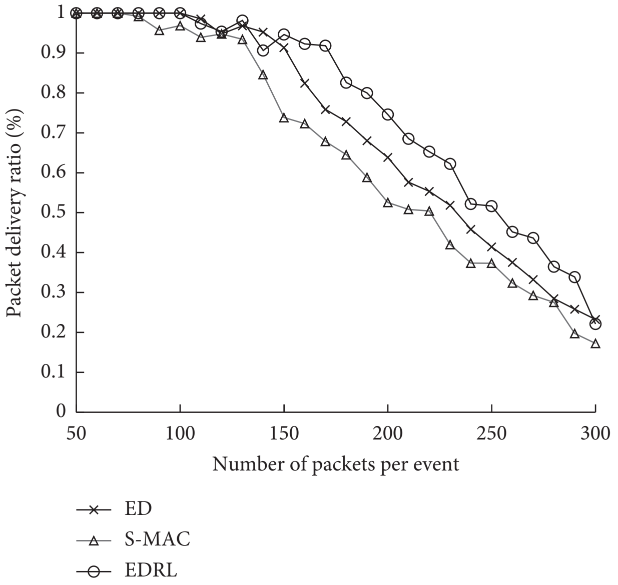


Figure 5: The comparison of delivery ratios as the load varies.

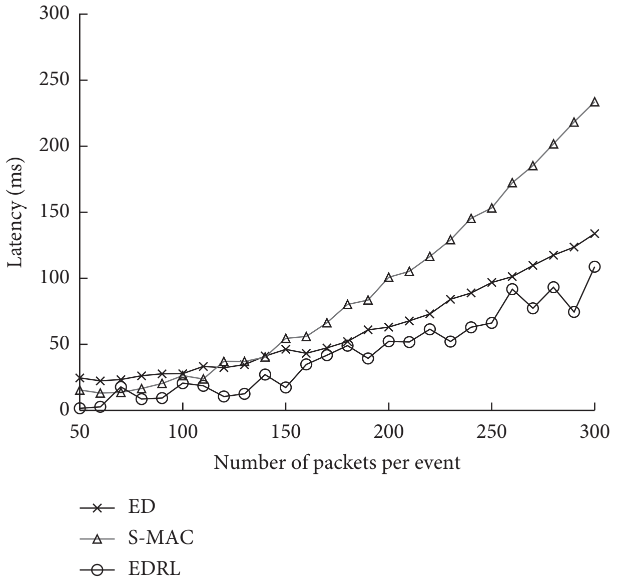


Figure 6: The comparison of end-to-end delays.

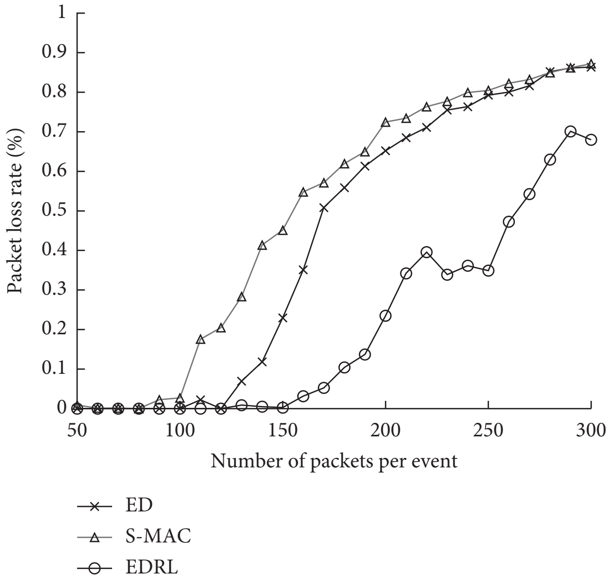


Figure 7: The comparison of packet loss rates.

number of packets per event exceeds 150. This is attributed to the reward function of RL in equation (14), which indicates that the path is decided based on latency. A packet is transmitted effectively by minimizing the number of hops to obtain the weight for each transmission schedule based on EDRL. Because of this, the proposed scheme is better than ED.

Figure 8 shows the average waiting time of the schemes. The proposed EDRL consistently outperforms the other schemes, which validates its effectiveness and robustness regardless of the load condition. It is achieved by properly selecting the node for packet transmission, which results in a reduced collision of the data transmission. The combination of RL with MC makes adaptive decisions results, which give more stable and better performance of the proposed scheme than S-MAC and ED. The proposed EDRL adjusts the duty cycle of the nodes in the multihop path according to the status of the network so that the waiting time incurred during packet transmission can be minimized.

Figure 9 shows the node survival rate of the three schemes. Observe from the figure that the fraction of dead nodes of the proposed scheme is substantially smaller than that of the other schemes. In the case of relatively low load, the fraction of dead nodes of ED scheme is smaller than that of S-MAC due to energy saving. As the load increases, the awakened nodes may increase transmission collision with the ED scheme, and as a result, the number of dead nodes becomes larger than S-MAC of fixed duty cycle.

Figure 10 shows the energy consumption rates of the three schemes. In the case of high traffic load, the transmission tasks and transmission conflicts significantly increase the energy consumption. The proposed EDRL scheme consistently outperforms the other schemes. This is attributed to the event-driven approach and RL, which reduce the operation time of the nodes and waiting time in forwarding the packet. It can be observed from the figure that the

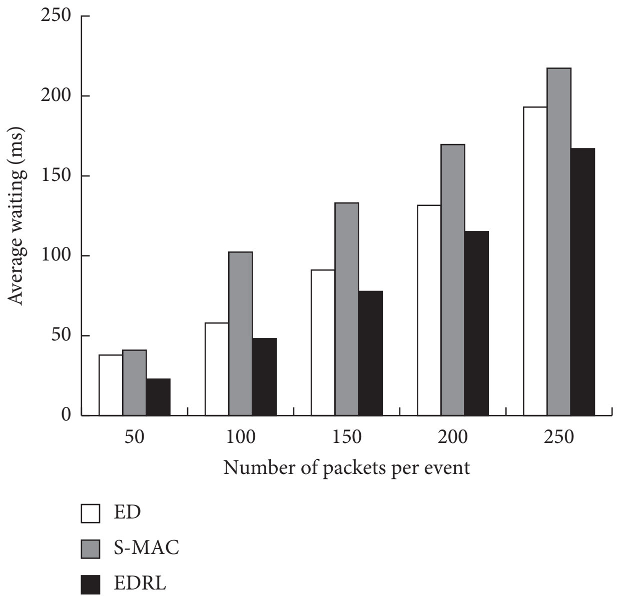


Figure 8: The comparison of average waiting times.

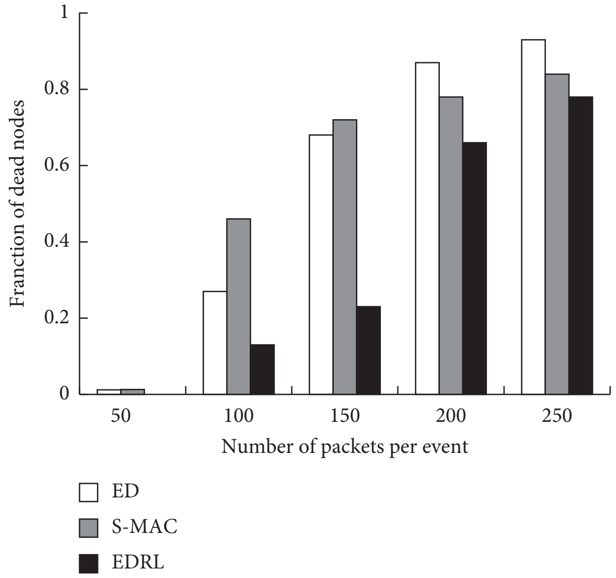


Figure 9: The comparison of the ratios of dead nodes.

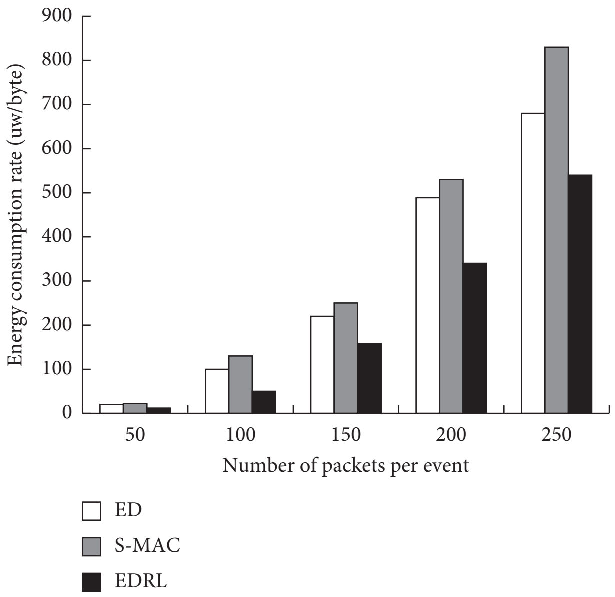


Figure 10: The comparison of energy consumption rates.

waiting time of the proposed EDRL scheme is significantly smaller than that of the other two schemes because of the set of nodes that is selected based on equation (5) and because it causes collision less likely.

5. Conclusion

In this study, an adaptive duty cycle scheduling scheme applicable to MAC has been proposed for WSN. It effectively improves the performance of the network by employing event-driven duty cycling and RL technique to effectively adapt to dynamic change in the network condition. In addition, the sampling approach based on Monte Carlo evaluation greatly improves the speed and accuracy for finding a suitable schedule. Computer simulation revealed that the proposed scheme substantially reduces the energy consumption, latency, and packet loss rate compared with the existing schemes.

In the future, we will further enhance the proposed scheme with a more sophisticated adaptive approach and reinforcement learning technique, along with the study on other performance metrics including throughput. We will also consider different learning techniques such as hybrid or federated learning to effectively cope with various operating conditions of the network. Furthermore, the proposed scheme will be extended to be applied to the virtualized network environment such as software-defined networking.

Data Availability

All data included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00133, Research on Edge computing via collective intelligence of hyperconnection IoT nodes), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute for Information and communications Technology Promotion) (2015-0-00914); Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2017R1A2B2009095, Research on SDN-based WSN Supporting Real-ftime Stream Data Processing and Multiconnectivity, 2019R1I1A1A01058780, Efficient Management of SDN-based Wireless Sensor Network Using Machine Learning Technique); and the second Brain Korea 21 PLUS project.

References

[1] A. Alemdar and M. Ibnkahla, "Wireless sensor networks: applications and challenges," in *Proceedings of the 9th International Symposium on Signal Processing and its Applications, 2007, ISSPA 2007*, pp. 1–6, IEEE, Sharjah, UAE, February 2007.

[2] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[3] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 171–180, ACM, Los Angeles CA, USA, November 2003.

[4] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," *Ad Hoc Networks*, vol. 3, no. 1, pp. 51–68, 2005.

[5] R. Zheng, J. C. Hou, and L. Sha, "June). Asynchronous wakeup for ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 35–45, ACM, Annapolis, MD, USA, June, 2003.

[6] F. Wang and J. Liu, "On reliable broadcast in low duty-cycle wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 767–779, 2011.

[7] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium, 2004*, p. 224, IEEE, Santa Fe, NM, USA, April 2004.

[8] J. J Niu, "Self-learning scheduling approach for wireless sensor network,"vol. 3, pp. V3–V253, in *Proceedings of the 2010 2nd International Conference on Future Computer and Communication (ICFCC)*, vol. 3, pp. V3–V253, IEEE, Wuhan, China, May 2010.

[9] M. Chincoli and A. Liotta, "Self-learning power control in wireless sensor networks," *Sensors*, vol. 18, no. 2, p. 375, 2018.

[10] S. Liu, K. W. Fan, and P. Sinha, "CMAC: an energy-efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, p. 29, 2009.

[11] Y. Lu, T. Zhang, E. He, and I. S. Comşa, "Self-learning-based data aggregation scheduling policy in wireless sensor networks," *Journal of Sensors*, vol. 2018, Article ID 9647593, 12 pages, 2018.

[12] N. Bouabdallah, M. E. Rivero-Angeles, and B. Sericola, "Continuous monitoring using event-driven reporting for cluster-based wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3460–3479, 2009.

[13] M. L. Littman, "Reinforcement learning improves behaviour from evaluative feedback," *Nature*, vol. 521, no. 7553, p. 445, 2015.

[14] S. Liu, G. Huang, J. Gui, T. Wang, and X. Li, "Energy-aware MAC protocol for data differentiated services in sensor-cloud computing," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–33, 2020.

[15] G. Li, F. Li, T. Wang, J. Gui, and S. Zhang, "Bi-adjusting duty cycle for green communications in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–55, 2020.

[16] S. Sundaresan, I. Koren, Z. Koren, and C. M. Krishna, "Event-driven adaptive duty-cycling in sensor networks," *International Journal of Sensor Networks*, vol. 6, no. 2, pp. 89–100, 2009.

[17] M. J. Miller and N. H. Vaidya, "Power save mechanisms for multi-hop wireless networks," in *Proceedings of the First International Conference on Broadband Networks, 2004 (BroadNets 2004)*, pp. 518–526, IEEE, San Jose, CA, USA, October 2004.

[18] Y.-C. Wang and J. M. Usher, "Application of reinforcement learning for agent-based production scheduling," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 1, pp. 73–82, 2005.

[19] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, pp. 1008–1014, 2000.

[20] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the International Conference on Machine Learning*, pp. 1889–1897, Lille, France, July 2015.

[21] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.

[22] H. Y. Huang, K. T. Kim, and H. Y. Youn, "Determining node duty cycle using Q-learning and linear regression for WSN," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 1–7, 2021.

[23] A. Soua and H. Afifi, "Adaptive data collection protocol using reinforcement learning for VANETs," in *Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1040–1045, IEEE, Cagliari, Italy, July 2013.

[24] J. Parras and S. Zazo, "Learning attack mechanisms in wireless sensor networks using markov decision processes," *Expert Systems with Applications*, vol. 122, pp. 376–387, 2019.

[25] F. Ren, J. Zhang, T. He, C. Lin, and S. K. D. Ren, "EBRP: energy-balanced routing protocol for data gathering in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2108–2125, 2011.

[26] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[27] S.-T. Cheng and T.-Y. Chang, "An adaptive learning scheme for load balancing with zone partition in multi-sink wireless sensor network," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9427–9434, 2012.

[28] Q. Chen, H. Gao, Z. Cai, L. Cheng, and J. Li, "Distributed low-latency data aggregation for duty-cycle wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2347–2360, 2018.